COMPARISON STUDY OF BLACK BOX AND WHITE BOX TESTING

Vibhu Chinmay
Student, B.Tech (VIth Sem), Electronics and Computers Engineering
Dronacharya College of Engineering, Gurgaon

Abstract: - Software testing is the process to uncover requirement, design and coding errors in the program. It is used to identify the correctness, completeness, security and quality of software products against a specification. Software testing is the process used to measure the quality of developed computer software. It exhibits all mistakes, errors and flaws in the developed software. Since the mid-1970s, software testing has been dominated by two major paradigms, known as black box testing and white box testing. Strategies for black box testing are based on the specification of the software component under test. Test suites developed using the black box paradigm reveal errors against the functional requirements of software components. Strategies for white box testing are based on the internals of software components. White box test suites are derived from the source code of the software component under test. Because programs can be represented in terms of graphs, solid coverage criteria can be defined for white box testing strategies. In this paper, we start with a brief overview of the principles underlying black box testing and white box testing. This fundamental problem in testing thus throws an open question, as to what would be the strategy we should adopt for testing. In our paper, we have described and compared the two most prevalent and commonly used software testing techniques for detecting errors, they are: white box testing and black box testing.

Index Term: Software Testing, Black Box Testing, and White Box Testing.

I. INTRODUCTION

Software testing is a most often used technique for verifying and validating the quality of software. Software testing is the procedure of executing a program or system with the intent of finding faults. It is measured to be labor intensive and expensive, which accounts for > 50 % of the total cost of software development. Software testing is a significant activity of the software development life cycle (SDLC). It helps in developing the confidence of a developer that a program does what it is intended to do so. In other words, we can say it’s a process of executing a program with intends to find errors. In the language of Verification and Validation (V&V), black box testing is often used for validation (i.e. are we building the right software?) and white box testing is often used for verification (i.e. are we building the software right?).This study emphasizes the need to investigate various testing techniques in software testing field; we have conducted a literature review to obtain the reviews from state-of-art.

Software testing identifies defects, flaws or errors in the application code that must be fixed. We can also define software testing as a process of accessing the functionality and correctness of a software through analysis. The main purpose of testing can be quality assurance, reliability estimation, validation and verification. Software testing is a fundamental component of software quality assurance and represents a review of specification, design and coding. The main objective of software testing is to affirm the quality of software system by systematically testing the software in carefully controlled circumstances, another objective is to identify the completeness and correctness of the software, and finally it uncovers undiscovered errors.

The two most important techniques that are used for finding errors are:

1) White Box Testing Technique: It is the detailed investigation of internal logic and structure of the code. In white box testing it is necessary for a tester to have full knowledge of source code.
2) **Black Box Testing Technique:** It is a technique of testing without having any knowledge of the internal working of the application. It only examines the fundamental aspects of the system and has no or little relevance with the internal logical structure of the system.

II. **WHITE BOX TESTING TECHNIQUE**

![Diagram of white box testing](image)

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. White box testing can uncover implementation errors such as poor key management by analyzing internal workings and structure of a piece of software. White box testing is applicable at integration, unit and system levels of the software testing process. In white box testing the tester needs to have a look inside the source code and find out which unit of code is behaving inappropriately.

Some of the advantages and disadvantages of white box testing technique are listed below:

**Advantages**

- It reveals error in hidden code by removing extra lines of code.
- Side effects are beneficial.
- Maximum coverage is attained during test scenario writing.

**Disadvantages**

- It is very expensive as it requires a skilled tester to perform it.
- Many paths will remain untested as it is very difficult to look into every nook and corner to find out hidden errors.
- Some of the codes omitted in the code could be missed out.

Some of the synonyms of white box testing are glass box testing, clear box testing, open box testing, transparent box testing, structural testing, logic driven testing and design based testing.

Some important types of white box testing techniques are briefly described below:

1) **Control Flow Testing:** It is a structural testing strategy that uses the program control flow as a model control flow and favors more but simpler paths over fewer but complicated path.

2) **Branch Testing:** Branch testing has the objective to test every option (true or false) on every control statement which also includes compound decision.

3) **Basis Path Testing:** Basis path testing allows the test case designer to produce a logical complexity measure of procedural design and then uses this measure as an approach for outlining a basic set of execution paths.

4) **Data Flow Testing:** In this type of testing the control flow graph is annotated with the information about how the program variables are define and used.

5) **Loop Testing:** It exclusively focuses on the validity of loop construct.

III. **BLACK BOX TESTING TECHNIQUE**
Black box testing treats the software as a “Black Box” – without any knowledge of internal working and it only examines the fundamental aspects of the system. While performing black box test, a tester must know the system architecture and will not have access to the source code.

Some of the advantages and disadvantages of black box testing technique are listed below:

**Advantages**

- Efficient for large code segment.
- Tester perception is very simple.
- Users perspective are clearly separated from developers perspective (programmer and tester are independent of each other).
- Quicker test case development.

**Disadvantages**

- Only a selected number of test scenarios are actually performed. As a result, there is only limited coverage.
- Without clear specification test cases are difficult to design.
- Inefficient testing.

Some of the synonyms of black box testing technique are opaque testing, functional testing, close box testing, and behavioral testing.

Some important types of black box testing techniques are briefly described below:

1) **Equivalence Partitioning**: It can reduce the number of test cases, as it divides the input data of a software unit into partition of data from which test cases can be derived.

2) **Boundary Value Analysis**: It focuses more on testing at boundaries, or where the extreme boundary values are chosen. It includes minimum, maximum, just inside/outside boundaries, error values and typical values.

3) **Fuzzing**: Fuzz testing is used for finding implementation bugs, using malformed/semi-malformed data injection in an automated or semi-automated session.

4) **Cause-Effect Graph**: It is a testing technique, in which testing begins by creating a graph and establishing the relation between the effect and its causes. Identity, negation, logic OR and logic AND are the four basic symbols which expresses the interdependency between cause and effect.

5) **Orthogonal Array Testing**: OAT can be applied to problems in which the input domain is relatively small, but too large to accommodate exhaustive testing.

6) **All Pair Testing**: In all pair testing technique, test cases are designs to execute all possible discrete combinations of each pair of input parameters. Its
main objective is to have a set of test cases that covers all the pairs.

7) **State Transition Testing:** This type of testing is useful for testing state machine and also for navigation of graphical user interface.

V. FUTURE OF SOFTWARE TESTING

With the changing trends in the software industry, software testing too changes. The existing new technologies like Service Oriented Architecture (SOA), wireless technologies, mobile services etc. has opened new path to testing. Some of the changes which we will see in the industry over the next few years are listed below:

- Testers will provide light weight models that developers can run against their codes.
- Early review and modeling will exposes many ambiguous bugs.
- As in the future developer’s code is full of testability hooks, errors will be more detectable.
- Static analyzer (detection tools) will come in main stream.
- Useful matrices such as spec coverage, model coverage and code coverage drives the projects.
- Combinatorial tools will allow testers to prioritize their testing.
- The testers will provide visible and value added services throughout the software development process.
- Tester can develop test harnesses stubs and drivers written in and interacting with a variety of programmatic languages.
- Tomorrows’ tester will be professionally more educated, examine and accredited professional.In the near future we will see a shift towards new techniques and testing transformed business operations, the way people interact with the systems and information it provides, and therefore mitigating the risk and increasing the benefits of business change.

VI. CONCLUSION

We can define software testing as an activity aimed at evaluating an attribute, or capability of a program to determine, that it meets its required specification. Software testing can provide an independent view of the software to allow the business to appreciate and understand the risk of software implementation.

To carry out software testing in a more effective manner, in our paper we have described and compared two main software testing techniques.

REFERENCES


